

WEST Search History

DATE: Thursday, April 10, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L5	l3[ti,ab]	4	L5
L4	L3 same (generic\$ or (platform near2 independent\$))	5	L4
L3	(modul\$ or applet or java) same cach\$ same (class\$ or categor\$ or root\$)	263	L3
L2	L1[ti,ab]	2	L2
L1	applet same cach\$ same (generic\$ or (platform near2 independent))	12	L1

END OF SEARCH HISTORY

WEST☐

L2: Entry 1 of 2

File: USPT

Feb 19, 2002

DOCUMENT-IDENTIFIER: US 6349333 B1

TITLE: Platform independent alarm service for manipulating managed objects in a distributed network management system

Abstract Text (1):

The client portion of a distributed client-server network management system uses an interface designed with, and written in, a platform independent language, such as Java. This allows management application programs to be written in the platform independent language and insures their portability. Further, in accordance with another aspect of the invention, the inventive interface allows a user to query an alarm log while the alarm log record information is cached at the server so that only a small client program is necessary. This program can run on a system with minimal resources, thereby allowing a network manager to manage a network from virtually any location on the network. For example, the client management application could be written as a Java applet which could then be run on any Java-enabled browser. In accordance with still another aspect of the invention, an alarm log can be queried on any user-defined attribute based on relational criteria such as: "greater than", "less than", "equal", "not equal", etc.

WEST**End of Result Set**

Generate Collection

Print

L2: Entry 2 of 2

File: USPT

Aug 28, 2001

DOCUMENT-IDENTIFIER: US 6282568 B1

TITLE: Platform independent distributed management system for manipulating managed objects in a network

Abstract Text (1):

The client portion of a distributed client-server network management system uses an interface designed with, and written in, a platform independent language, such as Java. This allows management application programs to be written in the platform independent language and insures their portability. The inventive interface caches most of the management information at the server so that only a small client program is necessary. This program can run on a system with minimal resources, thereby allowing a network manager to manage a network from virtually any location on the network. For example, the client management application could be written as a Java applet which could then be run on any Java-enabled browser. The inventive distributed management system can also connect with, and use, the services and structure of existing distributed network management systems.

WEST☐ Generate Collection☐ Print

L4: Entry 4 of 5

File: USPT

Sep 8, 1998

DOCUMENT-IDENTIFIER: US 5805796 A

TITLE: System architecture for implementing modular diagnostics

Detailed Description Text (32):

The generic access base class 640 is an abstract class that is a base class for generic access classes. The generic access classes are used by the device modules 330 to access computer hardware in a generic manner. In one embodiment of the software system 200, the general access base classes 640 either access "AT compatible" hardware directly or call the system access base classes 650 to access specific, non-AT compatible hardware such as cache/memory controllers, ECP parallel ports, CPUs and the like.

WEST☐ **Generate Collection** **Print**

L5: Entry 1 of 4

File: USPT

Aug 6, 2002

DOCUMENT-IDENTIFIER: US 6430564 B1

TITLE: Java data manager for embedded device

Abstract Text (1):

A data manager manages global data within a Java Virtual Machine (JVM) installed and running in an embedded device. The data manager maintains a data class list that stores data class identifiers associated with each data class object currently loaded and cached in a data cache in the embedded device. The data manager specifies a universal data manager API, which includes a data class loader method, a data object creation method, get and put data methods which allow manipulation of existing data objects, and an unload method which unloads cached data objects from the embedded device. The data manager comprises or is responsive to a memory management handler which detects low- or out-of-memory conditions and which selects one or more data class objects to be unloaded from the data cache.

WEST**End of Result Set**

Generate Collection

Print

L5: Entry 4 of 4

File: USPT

Jul 30, 1996

DOCUMENT-IDENTIFIER: US 5542078 A

**** See image for Certificate of Correction ****

TITLE: Object oriented data store integration environment for integration of object oriented databases and non-object oriented data facilities

Abstract Text (1):

A method and apparatus for accessing and effectively integrating non-object oriented data stores with object applications. An integrating environment is implemented wherein an application using a distributed object database and object database management system (ODBMS) is provided with an interface to external data stores in a manner so as to effect location transparency. The application, accessing data via the ODBMS, can manipulate data in foreign data stores which include external data that is mapped and converted into objects for use by object applications. A storage management application program interface ("SM API"), effects a functional interface for handling objects, referencing objects, implementing iteration and indexing of objects, and implementing object transaction and cache handling. The SM API is part of a modular architecture that includes an external storage manager which implements classes that provide the foundation for engaging external data stores, and which maps and converts external data into objects that can be manipulated by an application using the ODBMS.